利用Clob和Blob接口处理大数据对象

1. **大数据对象**主要有接口Clob和Blob的实现对象，在Clob中可以存储大字符数据对象，比如长篇小说，在Blob中可以存放**二进制大数据对象**，比如图片、电影、音乐等多媒体数据。
2. **Clob和Blob接口**都存在于java.sql包中，Clob是Character Large Object 的简写，Blob是Binary Large Object的简写。
3. （1）MySql中没有**Clob**这个数据类型，JDBC中的Clob对应的数据类型是TEXT类型。又分为LongText、MediumText、Text、TinyText四类，区别就是大小不同。MySQL中存在**Blob**这个数据类型。

（2）MySql中存在Blob数据类型，有LongBlob、MediumBlob、Blob、TinyBlob类型，区别是大小不同。

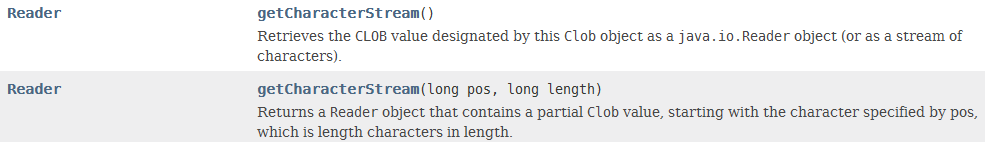
1. **Clob接口的方法：**
2. **length方法：获取长度。**



1. **getSubString方法**：获取**指定位置和指定长度**的字符串。



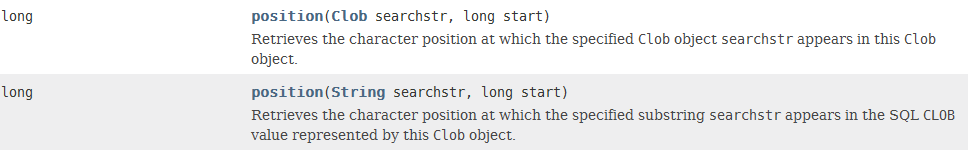
1. **getCharacterStream**：获取字符流。



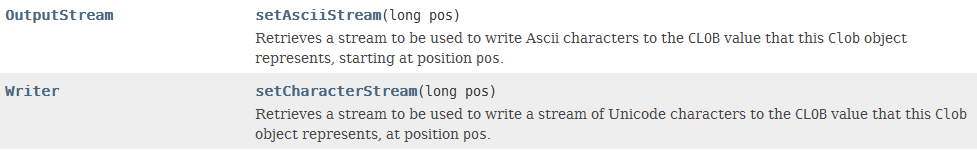
1. **getAsciiStream方法**：获取字节流。



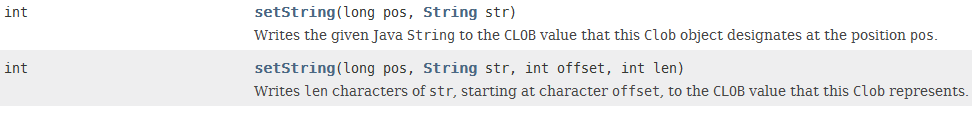
1. **position（）**：获取指针位置。



1. 添加**字节流和字符流**，从而给**Clob**继续添加内容。



1. 直接在Clob的指定位置添加**字符串**。



1. 截短方法：**truncate（long len）**

**truncate** [trʌŋ'keɪt; 'trʌŋ-] [想启用英文朗读功能吗？请先安装flash插件！](http://www.adobe.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash)vt. 把…截短；缩短；使成平面adj. 截短的；被删节的



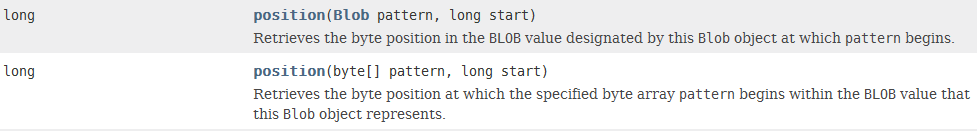
1. **Blob接口的方法：**
2. **length（）方法：**



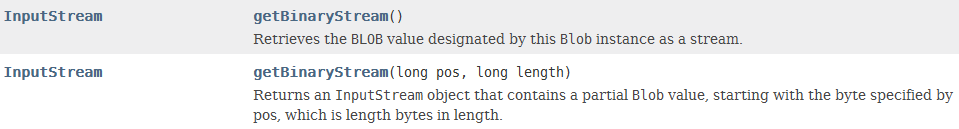
1. 获取字节数组：getBytes()



1. 获取指针位置：position()



1. 获取字节输入流：getBinaryStream()



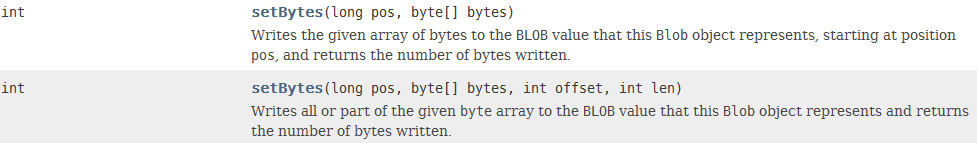
1. free：释放Blob对象和其他资源。



1. 通过字节输出流添加内容：setBinaryStream



1. 添加字节内容：setBytes（long pos，byte[] bytes）

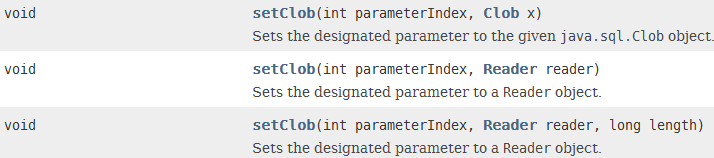


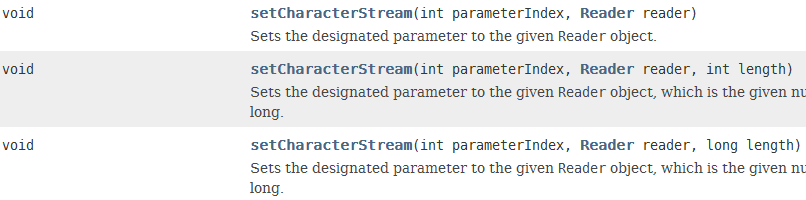
1. 截短方法：truncate（long len）



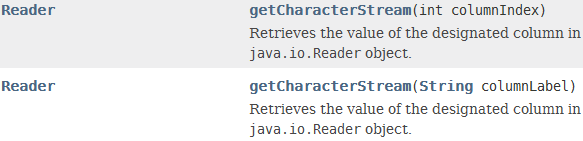
1. 
2. PreparedStatement的setClob方法或SetCharacterStream()方法：

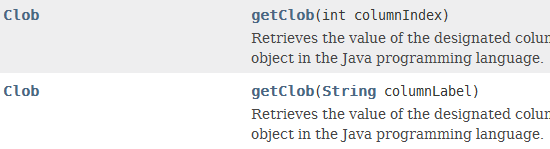
参数可以为**Clob对象或字符流Reader**。





1. PreparedStatement的getClob或getCharacterStream方法：





1. **处理Clob数据**：

**总结：PreparedStatement可以利用setClob（）或setCharacterStream（）方法。同理，ResultSet可以利用getClob（）或getCharacterStream（）方法。**

Clob在MySQl中对应的数据类型为longtext。

示例：（1）利用PreparedStatement的**setClob方法或SetCharacterStream添加Clob**：

DbUtil du = **new** DbUtil();//自定义的获取数据库连接的工具类

File contextFile = **new** File(“e:\\text2.txt”);

Book book = **new** Book(“山东科技大学校史2”,55,”诸葛福民2”,7,contextFile);

Connection con = du.getConnection();

String sql = “insert into t\_book values(null,?,?,?,?,?)”;

PreparedStatement psm = con.prepareStatement(sql);

psm.setString(1,book.getBookName());

psm.setInt(2,book.getPrice());

psm.setString(3,book.getAuthor());

psm.setInt(4,book.getBookId());

File file = book.getContextFile();

FileReader fr = **new** FileReader(file);

**psm.setClob(5,fr);**

**int** num = psm.executeUpdate();

du.close(con,psm);

例子2：

Connection conn = DriverUtil.*getConnection*();

String sql = “insert into bigtext(data) values(?)”;

PreparedStatement psm = conn.prepareStatement(sql);

File file = **new** File(“e:\\美文.txt”);

Reader reader = **new** BufferedReader(**new** FileReader(file));

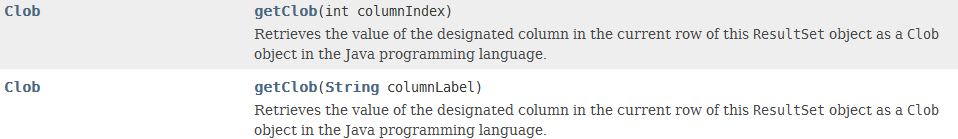
psm.setClob(1,reader,file.length());

// psm.setCharacterStream(1,reader,file.length());

**int** I = psm.executeUpdate();

psm.close(); conn.close();

（2）利用**ResultSet的getClob获取Clob，然后获取输出流**：或直接获取CharacterStream流。



1. 简单利用getSubString方法：

DbUtil du = new DbUtil();

Connection con = du.getConnection();

String sql = “select \* from t\_book “;

PreparedStatement psm = con.prepareStatement(sql);

ResultSet rs = psm.executeQuery();

while (rs.next()) {

**Clob clob = rs.getClob(“context”);**

**String str = clob.getSubString(1, (int) clob.length());**

System.out.println(str);

}

du.close(con, psm);

1. 利用**读取流**并使用写入流写到指定文件中去

DbUtil du = new DbUtil();

Connection con = du.getConnection();

String sql = “select \* from t\_book “;

PreparedStatement psm = con.prepareStatement(sql);

ResultSet rs = psm.executeQuery();

int I = 1;

while (rs.next()) {

Clob clob = rs.getClob(“context”);

**Reader reader = clob.****getCharacterStream();**

**FileWriter fw = new FileWriter(“e:\\texttext\\text\_read(“+i+”).txt”);**

char[] buf = new char[1024];

int len = 0;

while((len = reader.read(buf))!= -1) {

fw.write(buf,0,len);

}

reader.close();

fw.close();

i++;

}

du.close(con, psm);

3. Connection conn = DriverUtil.getConnection();

String sql = “select id,data from bigtext”;

PreparedStatement psm = conn.prepareStatement(sql);

ResultSet rs = psm.executeQuery();

File file = new File(“e:\\bbb\\aaaaa.txt”);

while(rs.next()) {

rs.absolute(9);

// Reader r = rs.getCharacterStream(“data”);

Reader r = rs.getClob(“data”).getCharacterStream();

BufferedWriter br = new BufferedWriter(new FileWriter(file));

char[] buff = new char[1024];

while(r.read(buff) != -1) {

br.write(buff, 0, buff.length);

}

br.close();r.close();

rs.afterLast();

}

rs.close();psm.close();conn.close();

1. **处理Blob数据**：

**psm可以利用setBlob（）方法或利用setBinaryStream（）方法。**

**同样ResultSet可以利用getBlob（）或getBinaryStream（）方法。**

1. 利用setBlob添加：

DbUtil du = new DbUtil();//自定义的类工具

File contextFile = new File(“e:\\text.txt”);

File music = new File(“e:\\111.mp3”);

Book book = new Book(“山东科技大学校史”,55,”诸葛福民”,7,contextFile,music);

Connection con = du.getConnection();

String sql = “insert into t\_book values(null,?,?,?,?,?,?)”;

PreparedStatement psm = con.prepareStatement(sql);

psm.setString(1,book.getBookName());

psm.setInt(2,book.getPrice());

psm.setString(3,book.getAuthor());

psm.setInt(4,book.getBookId());

**File file = book.getContextFile();**

**InputStream is = new FileInputStream(file);**

**psm.setBinaryStream(5,is,file.length());**

File musicFile = book.getMusic();

FileInputStream fis = new FileInputStream(musicFile);

psm.setBlob(6,fis);

int num = psm.executeUpdate();

du.close(con,psm);

if(num != 0) {

System.out.println(“添加成功”);

}else {

System.out.println(“添加失败”);

}

}

1. **利用getBlob获取**：

DbUtil du = new DbUtil();

Connection con = du.getConnection();

String sql = “select \* from t\_book “;

PreparedStatement psm = con.prepareStatement(sql);

ResultSet rs = psm.executeQuery();

int I = 1;

while (rs.next()) {

**Blob blob = rs.getBlob(“music”);**

**InputStream is = blob.getBinaryStream();**

**FileOutputStream fw = new FileOutputStream(“e:\\mp3mp3\\text\_read(“+i+”).mp3”);**

**byte[] buf = new byte[1024];//创建缓冲区**

int len = 0;

while((len = is.read(buf))!= -1) {

fw.write(buf,0,len);

}

is.close();

fw.close();

i++;

}

du.close(con, psm);

3. 例子：

Connection conn = DriverUtil.getConnection();

String sql = "select blobt from blobtest";

PreparedStatement psm = conn.prepareStatement(sql);

File file = new File("e:\\5454353464.jpg");

if(!file.exists()) {

file.createNewFile();

}

System.out.println(file.exists());

OutputStream os = new BufferedOutputStream(new FileOutputStream(file));

ResultSet rs = psm.executeQuery();

byte[] buf = new byte[1024];

while(rs.next()) {

InputStream in = rs.getBinaryStream("blobt");

for(int len = in.read(buf);len != -1;len = in.read(buf)) {

os.write(buf,0,len);

}

os.close();in.close();rs.afterLast();

}

rs.close(); conn.close();